PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) Internati nal Patent Classification 7 :
H04N 5/00, G06F 13/10

A1

(11) International Publication Number: WO 00/40007

(43) International Publication Date: 6 July 2000 (06.07.00)

(21) International Application Number: PCT/US99/30250

(22) International Filing Date: 17 December 1999 (17.12.99)

(30) Priority Data:
09/219,011 23 December 1998 (23.12.98) US

(71) Applicant: POWERTV, INC. [US/US]; Suite 100, 20833 Steven Creek Boulevard, Cupertino, CA 95014-2154 (US).

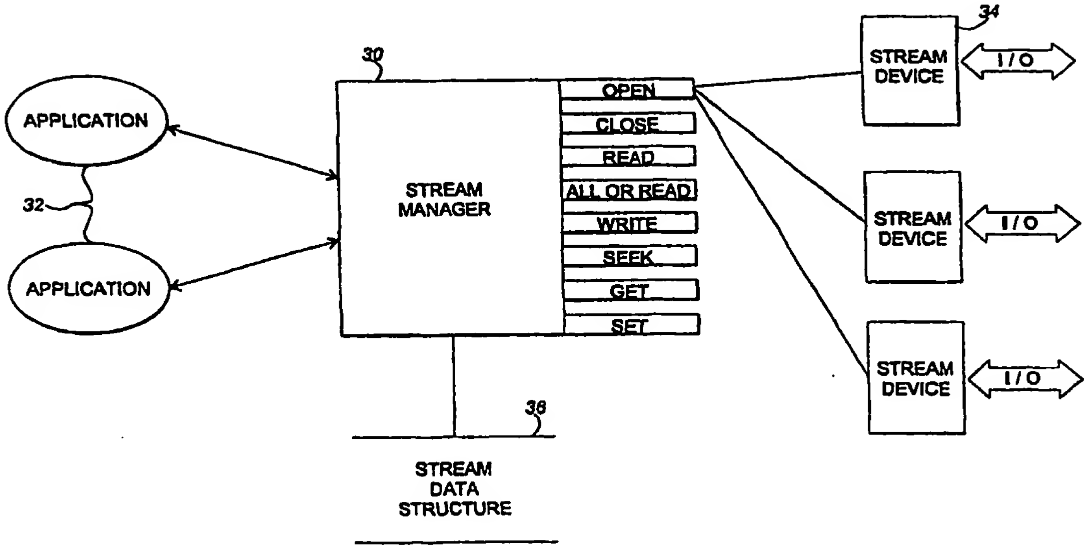
(72) Inventors: ASPROMONTE, Bill, J.; 1250 Glen Canyon Road, Santa Cruz, CA 95060 (US). STALKER, Altan, J.; 6315 Mayo Drive, San Jose, CA 95123 (US).

(74) Agents: MASSARONI, Kenneth, M. et al.; Scientific-Atlanta, Inc., Intellectual Property Dept., One Technology Parkway South, Norcross, GA 30092 (US).

(81) Designated States: KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: A STREAM DEVICE MANAGEMENT SYSTEM FOR MULTIMEDIA CLIENTS IN A BROADCAST NETWORK ARCHITECTURE



(57) Abstract

A stream device management system is provided for supporting applications that access a variety of stream devices associated with a conventional set-top box. More specifically, the stream device management system includes a stream manager configured to identify a plurality of stream devices and to store a device identifier for each of these stream devices, and a shared memory for storing stream data associated with each of the stream devices. To initiate communication with a first stream device, a first application sends a device identifier indicative of the first stream device to the stream manager. In response to receiving the device identifier, the stream manager communicates an address for the shared memory associated with the first stream device to the first application. Lastly, the application uses this address to access the stream data.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

A STREAM DEVICE MANAGEMENT SYSTEM FOR MULTIMEDIA CLIENTS IN A BROADCAST NETWORK ARCHITECTURE

BACKGROUND OF THE INVENTION

The present invention relates generally to an operating system for a multimedia client, and more particularly to a stream device management system for supporting applications that interact with a plurality of stream devices via a stream manager in a network architecture.

The Federal Communications Commission (FCC) has promulgated a new set of television broadcasting standards for implementing digital television (DTV). It is expected that digital television will eventually replace the current NTSC standard in homes across the country and throughout the world. Digital TV offers a number of advantages over the conventional NTSC standard, including high-definition pictures, vastly improved user interactivity, support for Internet email resources and the like.

We anticipate that the new digital TV standard will promote the confluence of computer and television technologies with far reaching impact. Thus, the personal computer will someday function as home entertainment multimedia system and the television will function as a node on a variety of computer networks ranging from the home automation network to the Internet. While there are a number of different possible physical packages for deploying digital television technology, currently most popular is the set-top box.

In the early days, the set-top box served primarily as the tuner and signal de-scrambler for cable TV. More recently, companies such as Scientific Atlanta have developed far more sophisticated set-top boxes. Interactive digital set-top boxes provide an open platform for developing and delivering interactive services and multimedia content to consumers across a broadcast network. To provide the power of the personal computer, these sophisticated set-top boxes incorporate a full-fledged computer operating system. Requirements such as management of high data throughput, effective management of memory in a constrained consumer device, and support for a secure environment drive the decision to architect an innovative operating system for use in set-top boxes.

In a set-top box environment, the operating system generally consists of layers of interconnected software modules designed to minimize redundancy and optimize multimedia processing. For instance, each hardware device associated with the set-top box is abstracted into a software device module. These software device modules are responsible for media delivery and manipulation of a particular hardware device (e.g., RAM, serial ports, SCSI hard drives, ect.) associated with the set-top box. Therefore, it is desirable that the operating system be designed for smooth and efficient porting between different hardware platforms. Thus, the operating system provides a device management system which serves as an interface between these software device modules and the applications residing on the set-top box.

In addition, the set-top box offers a challenging site to deploy a computer operating system. Memory resources with the set-top box are quite limited. Applications supported by the operating system must use memory in a highly efficient manner. Thus, it is also desirable that this device management system provide an efficient means for applications to use the typically small memory footprint associated with the set-top box.

SUMMARY OF THE INVENTION

In accordance with the teachings of the present invention, a stream device management system is provided for supporting applications that access a variety of stream devices associated with a conventional set-top box. More specifically, the stream device management system includes a stream manager configured to identify a plurality of stream devices and to store a device identifier for each of these stream devices, and a shared memory for storing stream data associated with each of the stream devices. To initiate communication with a first stream device, a first application sends a device identifier indicative of the first stream device to the stream manager. In response to receiving the device identifier, the stream manager communicates an address for the shared memory associated with the first stream device to the first application. Lastly, the application uses this address to access the stream data.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustrating the basic software components for an operating system in accordance with the present invention;

Figure 2 is a diagram showing a preferred embodiment of the stream device management system of the present invention;

Figure 3 is a flowchart showing how to initialize a stream device data structure associated with a stream manager of the present invention;

Figure 4 is a flowchart showing how to initiate communication between an application and a stream device using the stream manager of the present invention;

Figure 5 is a diagram illustrating a conventional Read operation between an application and a stream device; and

Figure 6 is a diagram illustrating an AllocRead operation between an application and a stream device in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The following description of the present invention is merely exemplary in nature and is in no way intended to limit the invention or its uses. Moreover, the following description, while depicting an operating system designed to reside on a conventional set-top box, is intended to adequately teach one skilled in the art to make and use an operating system for a variety of consumer multimedia clients including, but not limited to, intelligent televisions, Internet terminals and advanced DVD players.

Figure 1 depicts the basic software components of an operating system 10 in accordance with the present invention. The multitasking operating system 10 is designed to address the high-performance demands of media-centric, real-time applications being delivered through a set-top box. The operating system provides an open, scalable platform for developing and delivering multimedia content to consumers across broadcast and client/server networks. The software architecture for the operating system 10 is comprised from layers of interconnected modules designed to minimize redundancy and optimize multimedia processing in an interactive, network setting. The layers of the architecture include a hardware abstraction layer 12, a core layer 14, an application support layer 16 and an application layer 18.

Each hardware device associated with the multimedia client is abstracted into a software device module that resides in the hardware abstraction layer 12. Each of these device modules are responsible for media delivery and manipulation between a hardware device and the remainder of the operating system. Well defined application program interfaces (APIs) supported by each device module separate the hardware dependencies of each device from the portable operating system facilities, and thereby mask the idiosyncrasies of different hardware implementations.

A kernel and memory manager residing in the core layer 14 provide the base functionality needed to support an application. A fully preemptive, multithreaded, multitasking kernel is designed to optimize both set-top memory footprint and processing speed. Since the operating system will reside on consumer units, it has been designed to exist in a ROM-based system with a very small footprint (e.g., 1MB). In addition, the kernel has also been created to take advantage of 32-bit Reduced Instruction Set Computer (RISC) processors which enable high-speed transmission, manipulation and display of complex media types. On the other hand, a memory manager provides an efficient allocation scheme to enable the best performance from limited memory resources. Because embedded processors are likely to be the mainstay of consumer digital hardware implementations, the memory model requires little memory management unit support. The core layer 14 also provides an integrated event system and a standard set of ANSI C utility functions.

Built on top of the core layer 14 is an application support layer 16. This set of support modules provides higher-level processing functions and application services. Application management, session management, and tuner management are a few examples of these services. At the highest application level 18, at least one application, referred to as a resident application is always resident on a set-top box. The application level 18 also provides the necessary capabilities for authoring applications and for managing resources (e.g., the tuner) between the resident application and other background applications residing on the set-top box.

A device management system residing within the context of the application support layer 16 provides a consistent interface to the variety of stream devices supported by a typical set-top box. More specifically, a stream manager 30 facilitates communication between an application 32 and a stream device 34 as shown in Figure 2. For purpose of the following discussion, the term "application" signifies any software module, including the operating system, that may reside on the set-top box, and the term "stream" generally refers to any byte-oriented data (e.g. information in RAM or a file in an HTTP server) which may be accessed through a stream device 34. Although each stream device 34 has its own unique characteristics, the means for accessing these stream devices are similar. Therefore, stream manager 30 provides a standard set of APIs (e.g., read from a stream, write to a stream, etc.) to interact with each type of stream device 34. In this way, stream manager 30 is able to support several types of stream devices, including, but not limited to, memory, (SCSI) hard drives, serial ports, file transfer programs (FTP), hypertext transfer programs (HTTP), MPEG transport streams, and a broadcast file system (BFS). In addition, a stream data structure 36 accessible to the stream manager 30 is used to maintain a list of the valid stream devices associated with a particular multimedia client.

As previously discussed, the operating system 10 is comprised of a plurality of modules, including a software abstraction module for each stream device. During boot up of the operating system, each of the stream device modules are dynamically recognized and read into the stream data structure as shown in Figure 3. First, a loader module retrieves and examines every module associated with the operating system in block 42. To examine each module, block 46 parses an information string that contains data about each module. When decision block 48 detects a module that corresponds to a stream device (i.e., Group ID = "stream") it proceeds with initialization of the stream data structure; otherwise the next module is retrieved in block 42. Once all of the modules have been retrieved, processing is terminated in exit block 44.

For each stream device module, block 50 determines a device type identifier (e.g., "ftp", "memory", "bfs", etc.) by parsing the remainder of the information string. A mapping between a device type id and the software device module that contains the code for the stream device is maintained in the stream data structure. However, before block 54 stores this mapping in the stream data structure, block 52 executes an initialization method for that stream device. The process is repeated until the loader module has evaluated each module.

At this point, stream manager 30 is operative to initiate communication between an application 32 and a stream device 34. Referring to Figure 4, stream manager 30 supports an Open API that allows applications to gain access to a stream device. When opening a stream in block 62, the requesting application provides an input parameter to stream manager. Rather than using a data structure, this input parameter is a text-based ASCII string in the form of a conventional URL (e.g., serial0:///bps=19200;data=7;parity=0;stop=2). Because a text-based message is used to open the stream, variations in parameters for different device types or additions of new device types are easily be supported by the stream manager 30. A URL is composed of a scheme part and a scheme specific part. In the present invention, the scheme maps to a device type (e.g., serial0:///) while the scheme specific part maps to other device specific information. Thus, the device type identifier is a standard scheme. By using a URL based approach, the present invention is able to support any Internet based protocol without significant changes in the architecture.

Next, block 64 parses the input parameter to determine the device type id as provided by the requesting application. Using the device type id, decision block 66 searches the stream data structure for the corresponding stream device module. For a known device type, block 68 retrieves the module id from the stream data structure; whereas for an unknown device type processing branches to a diagnostic or error routine. Using this retrieved module id (as well as the other device specific access information), stream manager 30 is then able to gain access to the requested stream device. An OPEN API supported by each stream device is called in block 72. It should be noted that each of the standard APIs (e.g., OPEN, WRITE, READ, etc.) supported by the stream devices have a well know offset within the stream device module. Thus, only a module id is needed by the stream manager to initially access a stream device.

However, since a stream device typically supports multiple streams and an application can access one or more streams on the same or different stream devices, stream manager 30 also needs a means for identifying the particular stream it is accessing in relation to a stream device. Therefore, in response to each open request, a unique stream instance id is generated by the requested stream device and then communicated back to stream manager 30. In this way, a device module id in combination with a stream instance id (referred to as a stream id) uniquely identifies each stream.

Lastly, stream manager 30 returns the stream id to the application in block 74. For all subsequent access to an open stream, the application will use the stream id when calling the stream manager. Rather than perform the above-described mapping process, stream manager 30 will also use the stream id to directly accesses the open stream and its corresponding stream device.

Once an application has established communication with a stream device, it may perform various data access operations. Figure 5 illustrates the processing steps associated with a typical READ function. Prompted by a request from an application, the operating system allocates an application buffer space 82 in a shared memory 80 for each read request made by an application 32. When the application performs the read operation, the stream device 34 copies the requested information from its device buffer space 84 to the previously allocated application buffer space 82, and then passes an address (i.e., pointer) for application buffer space 82 back through the operating system to the requesting application 32. Using this address, the application 32 can access the data in the application buffer space 82. This approach is followed regardless of the type of information being requested from the device.

In addition to this conventional READ method, stream manager 30 of the present invention also provides a more efficient means for accessing stream data from a stream device. An Allocate Read Only ("AllocReadOnly") function is provided by stream manager 30 as depicted in Figure 6. Rather than use the standard READ API, an application 32 can also call the AllocReadOnly API when it is acceptable to receive read-only access to the stream data. In response, stream manager 30 will interact with the appropriate stream device 34 to provide the requesting application an address (e.g. a handle or a pointer) for the requested stream data. To do so, the AllocReadOnly API returns a handle to the stream device's internal buffer 84. In some cases, this internal buffer might be created from direct memory access.

Since the stream data is only being read, not manipulated, by the application (e.g., font information), there is no need to make an additional copy of the stream data in memory. In this way, the AllocReadOnly operation eliminates the additional copy step associated with a typical READ operation, and thus reduces memory allocation within the set-top box. Using the returned address, the requesting application 32 can directly read the stream data from the device's buffer space in shared memory. Once done with the data, the application 32 is responsible for freeing the memory space. An additional feature of the ReadAllocOnly function is that it allows the stream device to satisfy multiple requests for the same data. As opposed to multiple applications each having copies of the same data, multiple requesting applications receive the same address and thus read the same data. Therefore, it is preferable to architect an application so that read-only access is acceptable.

The foregoing discloses and describes merely exemplary embodiments of the present invention. One skilled in the art will readily recognize from such discussion, and from the accompanying drawings and claims, that various changes, modifications and variations can be made therein without departing from the spirit and scope of the present invention.

WHAT IS CLAIMED IS:

1. A device management system for supporting applications that reside on a multimedia client, the applications interacting with a plurality of stream devices associated with the multimedia client, comprising:

a stream manager being configured to identify the plurality of stream devices and store a device identifier for each of said stream devices;

a first application being operative to initiate communication between a first stream device and said first application by sending a device identifier to said stream manager, said device identifier indicative of said first stream device; and

said stream manager being operative, in response to receiving a device identifier from said first application, to stream data between said first application and said first stream device.

2. The device management system of Claim 1 wherein said device identifier includes a device type identifier and device access information for said first stream device.

3. The device management system of Claim 1 wherein said device identifier is a Uniform Resource Locator (URL).

4. The device management system of Claim 1 further comprises a shared memory accessible to said first application and said first stream device for storing stream data, said stream manager being operative to communicate an address for said shared memory to said first application and said first application using said address to read the stream data.

5. The device management system of Claim 4 wherein said stream manager being operative to communicate said address for said shared memory to a second application and said second application using said address to read the stream data.

6. A method for accessing stream data from a stream device by an application residing on a multimedia client, comprising the steps of:

configuring a stream manager to identify a plurality of stream devices associated with the multimedia client and to store a device identifier for each of said stream devices;

requesting access to a first stream device by a first application residing on the multimedia client, said first application sending a first device identifier indicative of a first stream device to said stream manager;

determining an address for stream data associated with said first stream device by said stream manager in response to receiving said first device identifier from said first application, the stream data being stored in a shared memory on the multimedia client;

communicating said address from said stream manager to said first application; and

accessing the stream data by said first application using said address.

7. The method of Claim 6 wherein the step of requesting access to a first stream device further comprises using a device type identifier and device access information for said first stream device as said first device identifier.

8. The method of Claim 6 wherein the step of configuring a stream manager further comprises using a Uniform Resource Locator (URL) as said device identifier.

9. The method of Claim 6 further comprises the steps of communicating said address for said shared memory to a second application and accessing the stream data by said second application using said address.

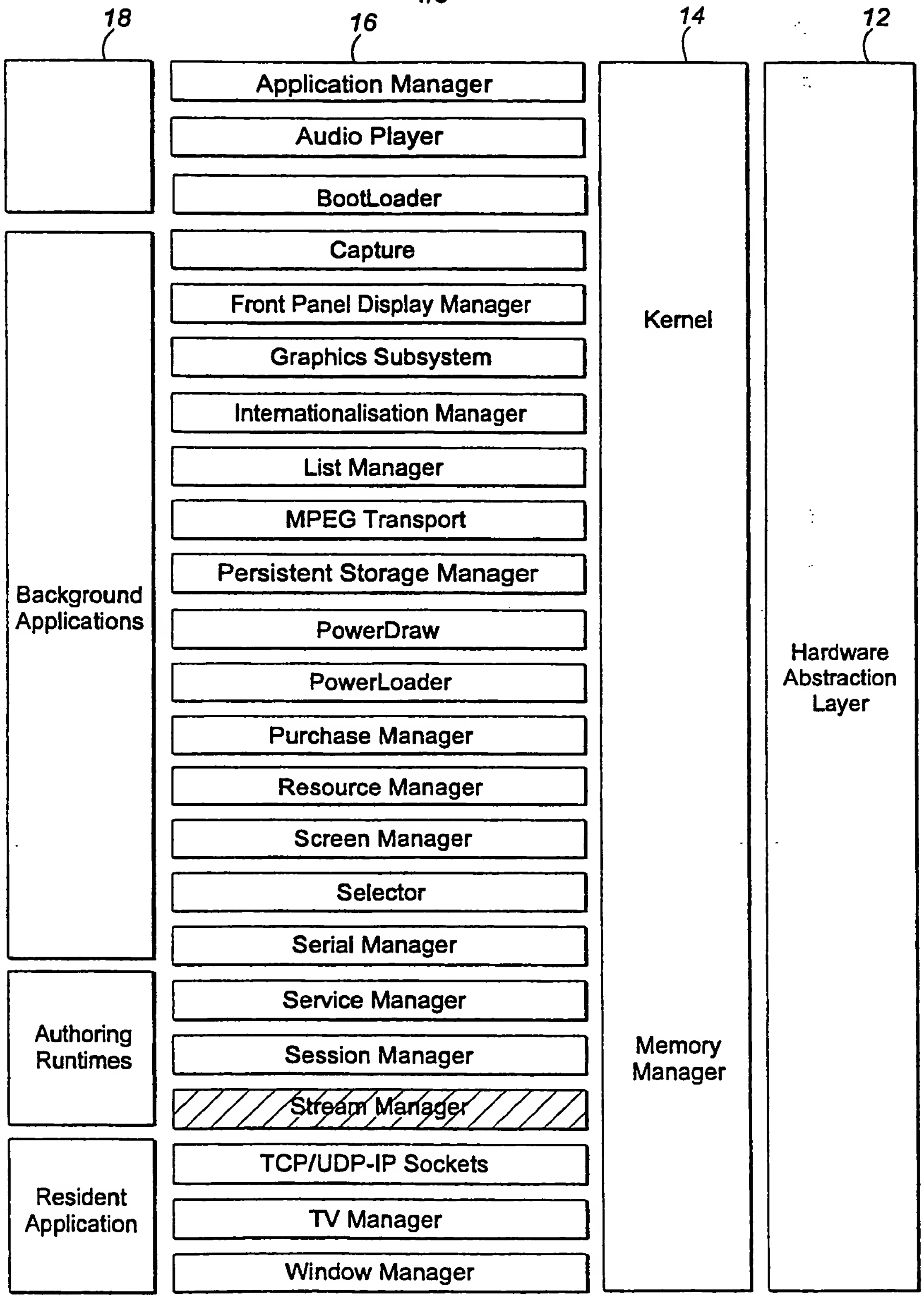

18
16
14
12
Application Manager
Audio Player
BootLoader
Capture
Front Panel Display Manager
Graphics Subsystem
Internationalisation Manager
List Manager
MPEG Transport
Persistent Storage Manager
PowerDraw
PowerLoader
Purchase Manager
Resource Manager
Screen Manager
Selector
Serial Manager
Service Manager
Session Manager
Stream Manager
TCP/UDP-IP Sockets
TV Manager
Window Manager
Background Applications
Authoring Runtimes
Resident Application
Kernel
Memory Manager
Hardware Abstraction Layer

FIG. 1

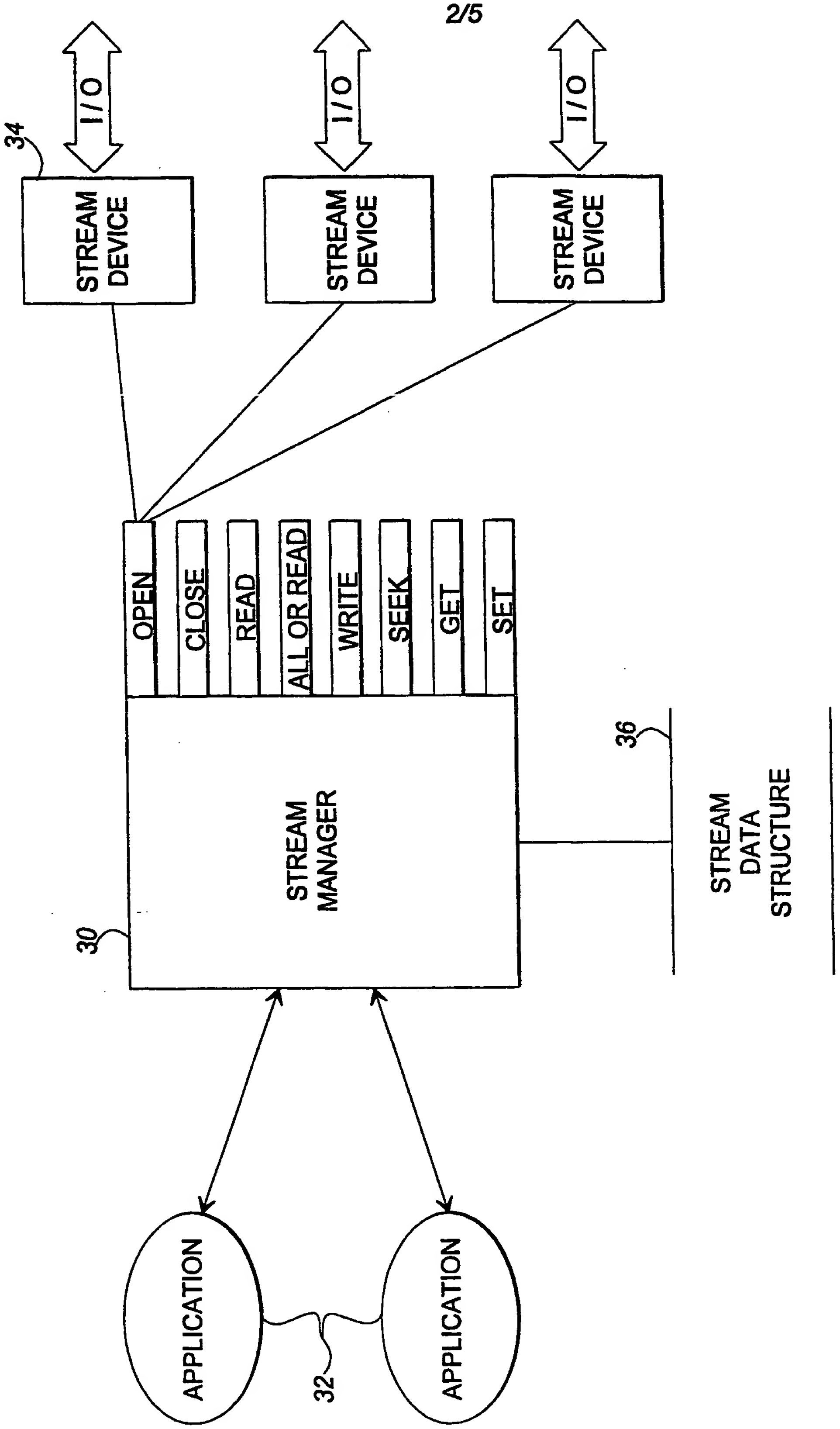

34
I/O
STREAM DEVICE
I/O
STREAM DEVICE
I/O
STREAM DEVICE
OPEN
CLOSE
READ
ALL OR READ
WRITE
SEEK
GET
SET
30
STREAM MANAGER
36
STREAM DATA STRUCTURE
APPLICATION
32
APPLICATION

FIG. 2

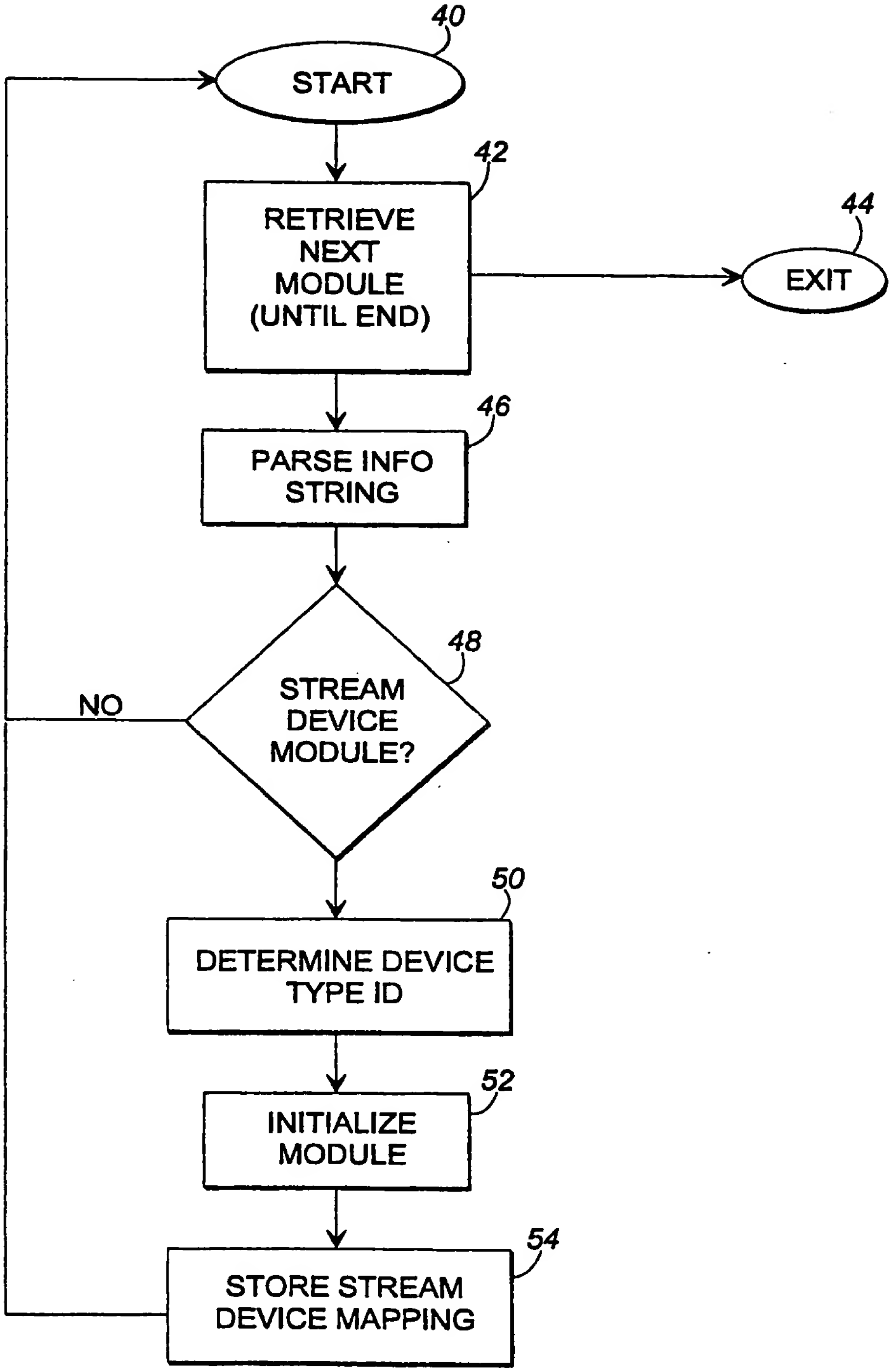

40
START
42
RETRIEVE NEXT MODULE (UNTIL END)
44
EXIT
46
PARSE INFO STRING
48
STREAM DEVICE MODULE?
NO
50
DETERMINE DEVICE TYPE ID
52
INITIALIZE MODULE
54
STORE STREAM DEVICE MAPPING

FIG. 3

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/30250

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N5/00 G06F13/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 43172 A (DECLERCK CHRISTOPHE ;SARFATI JEAN CLAUDE (FR); CANAL PLUS SA (FR);) 1 October 1998 (1998-10-01)	1,2,4-7, 9
Y	the whole document	3,8
Y	WO 97 49023 A (HANSON GORDON L) 24 December 1997 (1997-12-24) page 10, line 27 - line 33	3,8
A	abstract	1,6
X	EP 0 260 458 A (IBM) 23 March 1988 (1988-03-23) column 1, line 1 - line 41 column 3, line 37 -column 4, line 15 column 6, line 4 -column 7, line 10	1-3
A	abstract; figures 3,4	4-9
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 6 April 2000	Date of mailing of the international search report 12/04/2000
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer La, V

2

INTERNATIONAL SEARCH REPORT

Interna al Application No
PCT/US 99/30250

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 610 677 A (IBM) 17 August 1994 (1994-08-17) page 2, line 1 -page 3, line 15 page 3, line 45 - line 57 page 4, line 48 - line 58	1,4-6,9
A	abstract; figure 1	2,3,7,8
P,X	EP 0 946 053 A (CANAL PLUS SA) 29 September 1999 (1999-09-29) the whole document	1,4,6
A	EP 0 872 798 A (CANAL PLUS SA) 21 October 1998 (1998-10-21) abstract; figures 2,9	1-9
A	US 5 778 226 A (HOLMSTROM LARRY W ET AL) 7 July 1998 (1998-07-07) column 10, line 25 - line 67 abstract; figure 1	1-3,6-8

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interna al Application No
PCT/US 99/30250

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9843172 A	01-10-1998	AU 2638597 A	20-10-1998
		NO 994539 A	22-11-1999
		AU 2770697 A	20-10-1998
		AU 7038198 A	20-10-1998
		WO 9843425 A	01-10-1998
		WO 9843426 A	01-10-1998
		WO 9843162 A	01-10-1998
		WO 9843431 A	01-10-1998
		WO 9843248 A	01-10-1998
		WO 9843165 A	01-10-1998
		WO 9843415 A	01-10-1998
		WO 9843433 A	01-10-1998
		WO 9843427 A	01-10-1998
		WO 9843437 A	01-10-1998
		WO 9843167 A	01-10-1998
		WO 9843428 A	01-10-1998
		WO 9843421 A	01-10-1998
		EP 0872798 A	21-10-1998
		EP 0866611 A	23-09-1998
		EP 0866616 A	23-09-1998
		EP 0866613 A	23-09-1998
		EP 0968610 A	05-01-2000
		EP 0968609 A	05-01-2000
		EP 0968607 A	05-01-2000
		EP 0974229 A	26-01-2000
		EP 0974230 A	26-01-2000
		EP 0968468 A	05-01-2000
		EP 0968465 A	05-01-2000
		EP 0968602 A	05-01-2000
		EP 0968611 A	05-01-2000
		EP 0968608 A	05-01-2000
		EP 0968469 A	05-01-2000
		EP 0972406 A	19-01-2000
		NO 994529 A	19-11-1999
		NO 994530 A	19-11-1999
		NO 994531 A	19-11-1999
		NO 994532 A	19-11-1999
		NO 994533 A	22-11-1999
		NO 994534 A	22-11-1999
		NO 994535 A	22-11-1999
		NO 994536 A	22-11-1999
		NO 994537 A	22-11-1999
		NO 994538 A	22-11-1999
		NO 994540 A	22-11-1999
		NO 994541 A	22-11-1999
		NO 994542 A	22-11-1999
		NO 994543 A	22-11-1999
		NO 994544 A	22-11-1999
		ZA 9703605 A	10-09-1998
WO 9749023 A	24-12-1997	CA 2258596 A	24-12-1997
		EP 0974090 A	26-01-2000
EP 0260458 A	23-03-1988	DE 3788547 D	03-02-1994
		DE 3788547 T	30-06-1994
		JP 63163944 A	07-07-1988
EP 0610677 A	17-08-1994	JP 2557189 B	27-11-1996

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 99/30250

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0610677 A		JP 6301623 A	28-10-1994
		US 5671442 A	23-09-1997
		US 5535416 A	09-07-1996
EP 0946053 A	29-09-1999	AU 2953199 A	18-10-1999
		WO 9951021 A	07-10-1999
EP 0872798 A	21-10-1998	WO 9843248 A	01-10-1998
		AU 7038198 A	20-10-1998
		WO 9843167 A	01-10-1998
		EP 0968469 A	05-01-2000
		NO 994542 A	22-11-1999
		AU 2770697 A	20-10-1998
		WO 9843425 A	01-10-1998
		WO 9843426 A	01-10-1998
		WO 9843162 A	01-10-1998
		WO 9843431 A	01-10-1998
		WO 9843165 A	01-10-1998
		WO 9843415 A	01-10-1998
		WO 9843172 A	01-10-1998
		WO 9843433 A	01-10-1998
		WO 9843427 A	01-10-1998
		WO 9843437 A	01-10-1998
		WO 9843428 A	01-10-1998
		WO 9843421 A	01-10-1998
		EP 0866611 A	23-09-1998
		EP 0866616 A	23-09-1998
		EP 0866613 A	23-09-1998
		EP 0968610 A	05-01-2000
		EP 0968609 A	05-01-2000
		EP 0968607 A	05-01-2000
		EP 0974229 A	26-01-2000
		EP 0974230 A	26-01-2000
		EP 0968468 A	05-01-2000
		EP 0968465 A	05-01-2000
		EP 0968602 A	05-01-2000
		EP 0968611 A	05-01-2000
		EP 0968608 A	05-01-2000
		EP 0972406 A	19-01-2000
		NO 994529 A	19-11-1999
		NO 994530 A	19-11-1999
		NO 994531 A	19-11-1999
		NO 994532 A	19-11-1999
		NO 994533 A	22-11-1999
		NO 994534 A	22-11-1999
		NO 994535 A	22-11-1999
		NO 994536 A	22-11-1999
		NO 994537 A	22-11-1999
		NO 994538 A	22-11-1999
		NO 994539 A	22-11-1999
		NO 994540 A	22-11-1999
		NO 994541 A	22-11-1999
		NO 994543 A	22-11-1999
		NO 994544 A	22-11-1999
		ZA 9703605 A	10-09-1998
		ZA 9703606 A	07-09-1998
US 5778226 A	07-07-1998	CA 2010591 A,C	20-04-1991

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interna al Application No
PCT/US 99/30250

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5778226 A		US 5459867 A	17-10-1995